

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2012 Proceedings

Proceedings

Balanced Resource Allocation

Thomas Schuster

Forschungszentrum Informatik, Karlsruhe, Germany, schuster@fzi.de

Follow this and additional works at: <http://aisel.aisnet.org/amcis2012>

Recommended Citation

Schuster, Thomas, "Balanced Resource Allocation" (2012). *AMCIS 2012 Proceedings*. 14.
<http://aisel.aisnet.org/amcis2012/proceedings/SystemsAnalysis/14>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2012 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Balanced Resource Allocation

Thomas Schuster

FZI Forschungszentrum Informatik

Haid-und-Neu-Straße 10-14

76131 Karlsruhe, Germany

schuster@fzi.de

ABSTRACT

Resource management is a key issue in execution of business processes and tasks. While control flow structure is given by underlying business process models, assignment of process activities to resources is a runtime task that includes optimization questions. Thus reasoning about system optimization is based on precise specification of resources and tasks. However, resource models previously employed in business process management lacked preciseness to enable sound analysis and optimization. In this article a modeling method that contains necessary information will be utilized and formalized in order to pose and solve optimization questions. Furthermore assignment of human resources is combined with further education, thus an optimized resource assignment will foster balanced further education. In consequence performance peaks of single resources will be alleviated and risk of failure can be diminished.

Keywords

Resource Management, Human Resources, Resource Allocation, Business Process, Learning

INTRODUCTION

Shortened product and process life-cycles facilitate changes of resource requirements; considering task execution, especially capabilities of human resources are affected. Hence, the selection and allocation of resources (external business partners as well as intra organizational resources) and management of resource capabilities, as part of resource management, can be identified as keys to success (Du, Davis, Huang and Shan, 1999; Pfeffer and Salancik, 2003). This does include resource recruiting as well as planning of further education, if human resources are involved.

A lot of delays in task execution can be eliminated or reduced, if sound process and resource analysis or simulation is performed. In order to enable analysis it is necessary to create models that represent this system (Niemeyer, 1977), in this case usually business process and resource models are employed. Business process models allow analysis and improvements in the order of task execution, e.g. by identification of potential conflicts, deadlocks or detection of sections which could possibly be implemented concurrently. Analysis of resource models on the other hand can reveal if necessary resources are available or if the amount of resources is sufficient (Russell, van der Aalst, ter Hofstede and Edmond, 2005).

At runtime, the coordination of tasks and resources is a major job of workflow management systems (van der Aalst and Kumar, 2001; zur Mühlen, 2004; Hollingsworth, 2005). If, however, the structure of business processes is not known or cannot be changed, task allocation can also be considered being event-driven. In an event-driven view, for a given time step a certain amount of resources has to be allocated to a certain amount of tasks (Page and Kreutzer, 2005). Task execution can then be optimized through choice of allocation strategy; graph based analysis or linear optimization, for instance, can improve resource allocation for a given objective.

The remainder of this article is organized as follows: The next section provides an overview on resource modeling; the Resource Modeling Language (RML) is introduced as technique to model resources and their properties. Furthermore an instance of RML and its formalized representation is given in preparation of resource analysis. In the following section an algorithm which enables balanced resource allocation – with focus on resource capabilities – is presented. Consecutively a case study, in which balanced resource allocation is evaluated in context of ticket management, does demonstrate practicability of theoretical concepts. Finally the paper is concluded by a summary and an outlook on ideas for future research.

RESOURCE MODELING

Resource models enact efficient management of resources. Often resources are independently managed by pre-existing systems. Therefore integrating information, managed by these systems, in resource model repositories is critical to global management of resources. A second aspect is actualization of resource models according to changes.

In business process management nomenclature resources are entities (such as materials, machines or employees) that can be assigned to a task (Hollingsworth, 2005; zur Mühlen, 1999). During runtime these resources have to be coordinated in order to enable execution of tasks and hence process execution. Resource management must prevent or handle concurrent access and access of temporally unavailable resources. Thus reasoning about business processes and resources requires an understanding of requirements (given by process models) and properties of resources. Especially knowledge of similarities and differences among resources influences process execution and fosters further insights (such as identification of critical resources or bottlenecks). Thus in resource management, resources with similar properties are grouped to resource classes. This idea has evolved within last decades (Gutenberg, 1951), while new class types (such as information) have been added according to industrial development (Kern, 1988; Staudt Lerner, Ninan, Osterweil and Podorozhny, 2000).

In order to enable reasoning relevant resources have to be comprised in a resource model. So as to allow sound analysis, a lot of but especially the following properties of resources have to be monitored and therefore somehow reflected in resource models:

- *Classes*: resources have to be distinguishable in order to enable allocation based on class types.
- *Current state*: of a resource of resources must be monitored and reflected by resource models. State usually includes a state type (such as *ready*) as well as degree of utilization.
- *Spatial and temporal constraints*: resources may be temporarily unavailable and can be distributed across several locations. Localization of the resources may be a critical aspect in scheduling; e.g. worldwide operating project teams face spatial distribution which influences temporal availability (Cook and Churcher 2005).
- *Capabilities*: allocation of resources can be capability-based; therefore capabilities of resources have to be represented in resource models. Especially for human resources capabilities are usually further distinguished in competences, skills and knowledge (Harzallah, Berio and Vernadat, 2006; Winterton, Delamare- Le Deist and Stringfellow, 2005).

Resource properties may alter over time. Capabilities, existence or availability of resources, for instance, may change during business process execution (e.g. a material is consumed and amount of this material is reduced). Also allocation of resources can be subject to rules (e.g. business rules or statutory provisions), compliance has to be ensured. This can be monitored and enforced by a resource service that manages resources according to rules and resource models (Adams, 2009).

Several approaches to model resources have been proposed so far (zur Mühlen, 1999; van der Aalst, Kumar and Verbeek, 2003), many of them are domain specific (e.g. Russell and van der Aalst, 2008), some are more general (e.g. Gutenberg, 1951). To enable allocation of resources some models in the context of business process modeling have been suggested (such as Oberweis and Schuster, 2010; Russell, van der Aalst, ter Hofstede and Edmond, 2005), however most of them are basic neglecting certain class types or properties. In order to support allocation principals suggested later in this article the Resource Modeling Language (RML, Oberweis and Schuster, 2010; Schuster and Weiß, 2011) will be employed to model resources. RML is based on a MOF-compliant (OMG, 2006) meta-model (RMM) which allows precise definitions of concepts and relationships (realized through abstract syntax definition and usage of Object Constraint Language, OCL, OMG, 2010). Furthermore this type of language definition enables automated generation of tool support by means of model driven software development (MDSD); hence changes in language definition can be easily reflected by model editors. RML consists of several model parts. The top-level part provides general resource descriptions which defines class types of resources. A specialization part focuses human resources and organizational structures (human resource meta-model, HRMM), reusable competence descriptions are added by the part competence meta-model (COMM).

Figure 1 illustrates the HRMM meta-model part. For sake of simplicity only key classes and relationships are displayed; several attributes, constraints (OCL), some classes and associations are omitted. Central resource class of HRMM is *HumanResource* which reflects manpower. Organizational structure is reflected by associations of resources to organizational units (*OrganizationUnit*), positions (*OrganizationalPosition*), groups (*OrganizationalGroup*) and roles (*OrganizationalRole*).

Furthermore the concepts of knowledge, skill and competence are explicitly modeled and associated to roles (*Role*) and human resources (*HumanResource*), thereby reflecting capabilities. While associations of these concepts to roles describe requirements, associations to human resources describe properties a resource. In the remainder of this article the concepts knowledge, skill and competence will be referred to as capabilities. In order describe temporal constraints (in terms of capacity) work time can be associated to human resources (during runtime this has to be monitored by a resource management component). In order to track current and past activities of human resources, projects and tasks as well as

participation of human resources can be modeled (in Figure 1 given by associations to *Project*, *TaskStatistics* and *Task*). Similar to constraints and capabilities this may alter over time and has to be monitored and updated by a resource management component.

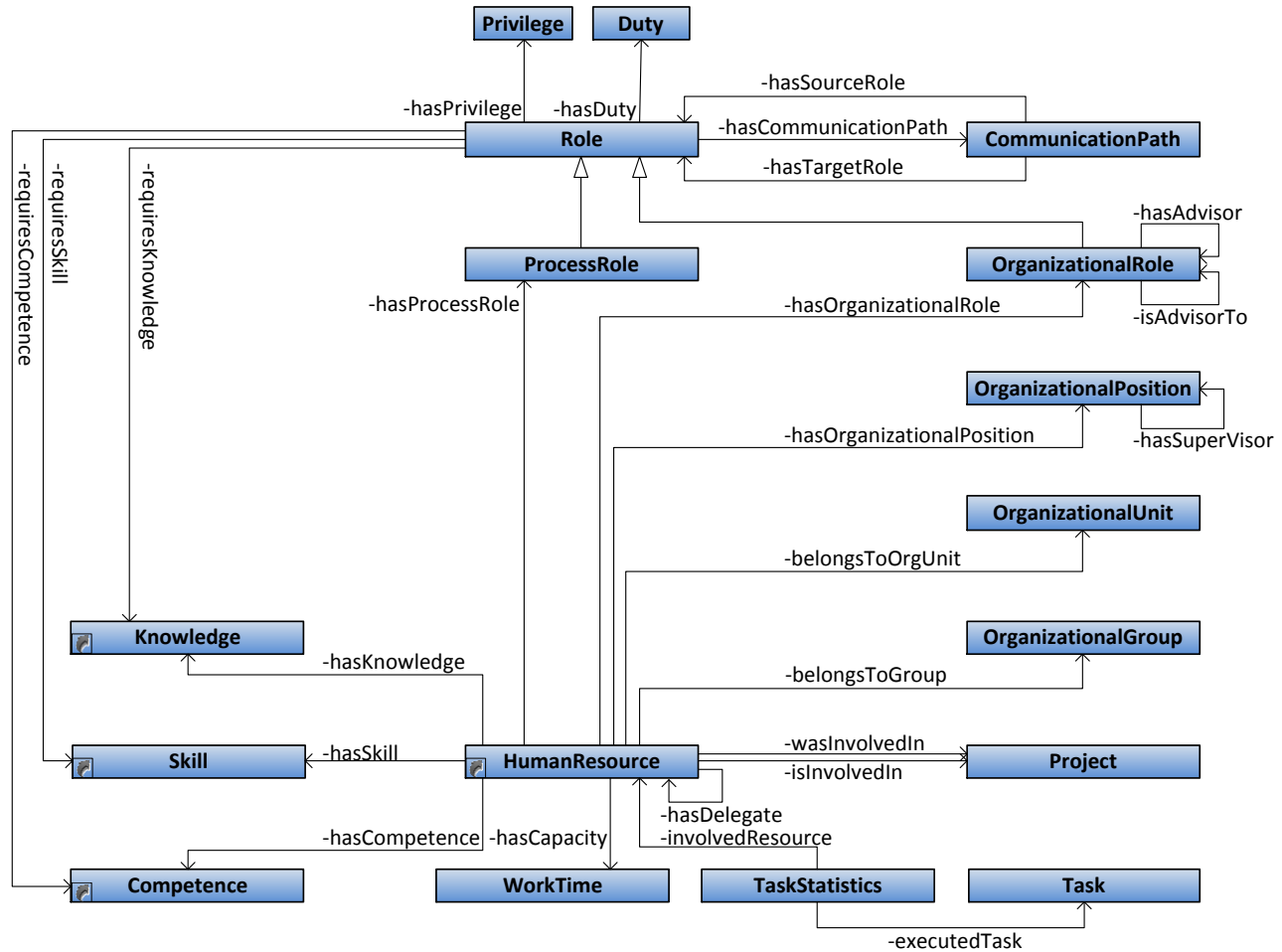


Figure 1. Resource Modeling Language (HRMM)

Resource Model

Figure 2 does display a resource model instance. As depicted a maintenance group inside an *IT Department* and its members are included in the model instance. The team consists of a manager, one senior developer and four developers; some further information is given by a property view of the model editor at the right hand side of Figure 2. A lot of information (such as contact information or projects) can be associated this way. Note that capabilities are only shown partially inside the property view, e.g. if a competence depends on two skills which their selves depend on several knowledge artifacts, then only the competence is shown if it is being possessed by a resource. In a subsequent step modeled RML information is beneficial for allocation of resources to tasks (see next and algorithm section of this article). The shown instance depicts a resource model which was made anonymous and has also been used in the case study of this article.

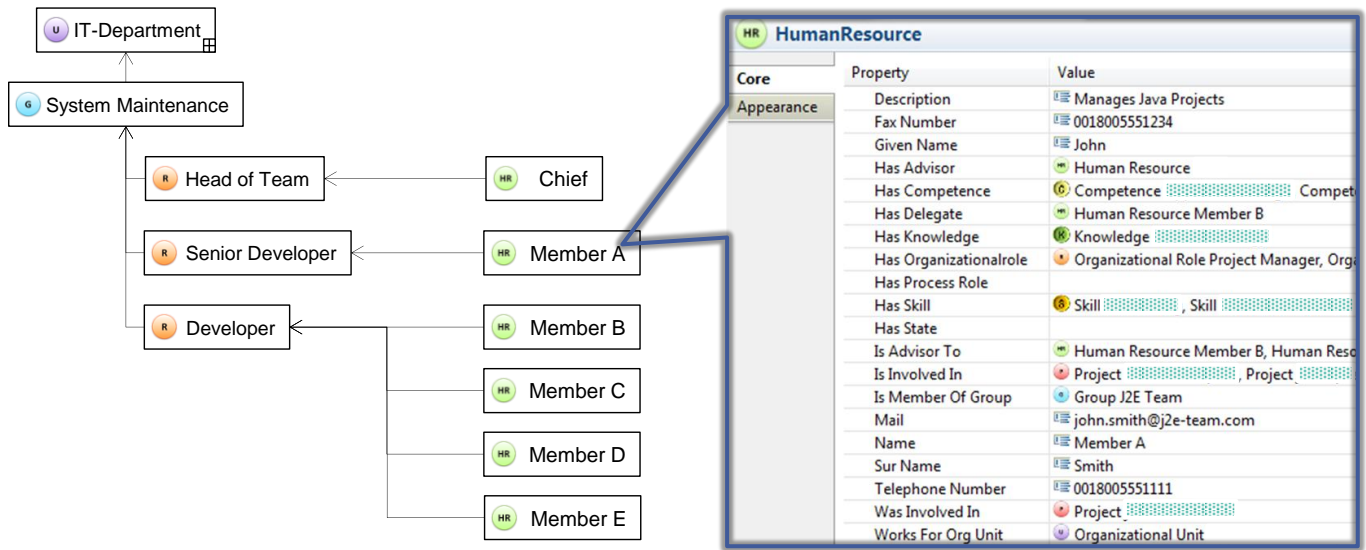


Figure 2. Instance of RML

Formal Model Representation

The meta-model given above can be formalized by the definition of a relation HR that does describe properties of given human resources. All properties of resources can be expressed as sets over which HR is defined. The abbreviation used inside property views of the modeling editor is used too, hence dependencies between capabilities are only shown by reusable capability models (in RML this would be COMM instances). Since allocation strategies given in the following algorithm and case study are capability based only, a shortened definition (just based on resources and capability concepts) of HR will be given here. Let

- R be a set of resources,
- $K = 9^n$ be a set of knowledge instances including level description,
- $S = 9^n$ be a set of skills including level description and
- $C = 9^n$ be a set of competences including level description,

then $HR \subseteq R \times K \times S \times C$. Properties of a single resource such as *Member A* given in Figure 2 may then be expressed as:

Domain	Properties of <i>Member A</i>
K	(0,0,0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0)
S	(0,0,3,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
C	(1,2,0,5,0,6,0,3)

Table 1

In short: (*Member A*, ($K_{13}:6$), ($S_2:3$), ($S_{11}:5$), ($S_{21}:4$), ($C_0:1$), ($C_1:2$), ($C_3:5$), ($C_5:6$), ($C_7:3$)). Due to abbreviated notation the capability model does reveal that Member A does possess a lot more knowledge and skill instances, e.g. ($K_1:5$) which is implied by ($S_{12}:6$), furthermore ($S_{12}:6$) is implied by ($C_3:5$). Building the reflexive transitive closure results in revealing all capability information in *HR*. Since this would be a lot of data for this model it is omitted. However, it is notable that this detailed representation is being utilized to compute capability distances.

ALLOCATION ALGORITHM

The algorithm (CBRA – Capability Balancing Resource Allocation) introduced in this section does optimize resource allocation in order to balance capability development of resources. As precondition of algorithm execution the modeling of capability and requirements profiles (De Coi, Herder, Koesling, Lofi, Olmedilla, Papapetrou and Sibershi, 2007) is necessary. While requirements profiles represent allocation constraints which need to be fulfilled for task execution, capability profiles associated to concrete resources reflect properties that can be matched to requirements profiles. CBRA consists of five steps of which the first three are preprocessing steps of given models. The first step is an optional step that does calculate importance of tasks. In a second step a weighted distance vector is calculated for each task-resource pair, which is consecutively mapped to a distance norm. In the fourth step a graph representation of possible task-resource pairs is generated and finally a so called minimum matching tree (MMT) is calculated.

Preprocessing Steps

The calculation of task importance is based on business process structure. Mainly this is done by a corrected version of the algorithm presented in (Fenz, Ekelhart and Neubauer, 2009). Since predefined process models have not been used within the following case study this step was not applicable and detailed presentation will be omitted. As first step in this case the distance vector $d_{ij} = \alpha_i \cdot v_{ij}$ that maps distance of task i to resource j is calculated. α_i describes task importance (in the case study α_i was set to one, since all tasks had to be considered equally important). Thus v_{ij} is given by requirements and capability profiles as:

$$v_{ijk} = \begin{cases} |A_{ik} - K_{jk}| \cdot c_k & K_{jk} > 0 \\ 8 \cdot c_k & K_{jk} = 0 \end{cases}$$

v_{ijk} is entry k of vector v_{ij} . c_k is a weight that can be associated to a reusable capability modeled in RML (COMM-Instance). A_{ik} does express the level of capability k required to execute task i . K_{jk} does reflect the actual level of capability k possessed by resource j . In case that a resource does not possess a capability at all ($K_{jk} = 0$), the distance entry is maximized. In this case 8 is the maximum value, because according to EQF (CEN, 2008) eight capability levels are defined in RML. In a following preparation step each task-resource relation expressed by a distance vector is transformed to a length measurement Φ_{ij} based on the Euclidean norm. Calculation of this length measurement has wide influence on allocation optimization. Therefore three variations are suggested. Variation 1 can be utilized to generate a best fit; variation 2 enforces choice of the most inexperienced resource, while variation 3 is a mixture between both other variations. Thus variation 2 fosters a balanced learning curve of allocated resources, while variation 3 favors further education of untrained resources. The parameter δ (threshold) can be used to adjust these strategies (through modification of Φ_{ij}). In conclusion the variations define different strategies for task allocation; choice and effects of each strategy is being discussed in the case study section below.

$$\Phi_{ij} = \|d_{ij}\|_2 \quad \Phi_{ij} = \begin{cases} 1 & \|d_{ij}\|_2 > \delta \\ \frac{1}{\|d_{ij}\|_2} & \delta \geq \|d_{ij}\|_2 > 1 \\ \frac{1}{1 + \|d_{ij}\|_2} & 1 \geq \|d_{ij}\|_2 > 0 \\ \frac{1}{k} & \|d_{ij}\|_2 = 0 \end{cases} \quad \Phi_{ij} = \begin{cases} \infty & \|d_{ij}\|_2 > \delta \\ \frac{1}{\|d_{ij}\|_2} & \delta \geq \|d_{ij}\|_2 > 0 \\ \max_{j \in HR} \frac{1}{\|d_{ij}\|_2} & \|d_{ij}\|_2 = 0 \end{cases}$$

Variation 1
Variation 2
Variation 3

Operational Steps

Let T be a period of time, $A = \{a_1, a_2, \dots, a_n\}$ a set of tasks that need to be allocated within T and $T_i = \{t_1, t_2, \dots, t_n\}$ a set of discrete points of time within T . If \mathcal{R} is a binary relation (A, T_i, \circ) with $(a_i, t_i) \in \mathcal{R}$ if a_i has to be allocated at t_i , then $A_t = \{a_i | a_i \mathcal{R} t\}$ is the set of tasks to be allocated at time t . $R = \{r_1, r_2, \dots, r_n\}$ is a set of available resources.

In respect to the preprocessing steps an allocation problem at time t_i can be expressed as weighted graph $G = (V, E, f)$ in which tasks and available resources are mapped to vertices (precisely this would be two types of vertices) and edges connect task to resource vertices. $f: E \rightarrow \mathbb{R}$ is a function that associates each edge with a corresponding distance measure calculated in the preprocessing steps, which means that $f(e_{ij}) = \Phi_{ij}$. Note that if variation 3 is chosen as distance measure, then

distances calculated with ∞ are not represented by edges in G . In order to find an allocation of resources – for the execution of each task – at minimum distance amongst requirements and actual capabilities a perfect matching at minimal weight could be computed (e.g. by the algorithm suggested by Hopcroft and Karp, 1971). In consideration of runtime aspects this problem is now transformed to the search of a tree similar to a minimum spanning tree (MST) – the computed tree will be referred to as minimum matching tree (MMT). This search will be resolved by a modification of Borůvka's algorithm (Borůvka, 1960). In order to conduct this problem conversion the graph is extended by a single vertex W , which will be the root of the calculated tree. Furthermore, for each task $a_i \in A$ an edge to W with weight 0 is added to G .

Hence the distance measure Φ_{ij} can be represented in a matrix in which rows correspond to resources available at time t_i and columns correspond to tasks to be allocated at time t_i . An example is given in the matrix below, which is also represented by a graph G in Figure 3. A transformation of the allocation problem such that a MMT can be calculated is depicted by Figure 4.

$$M_{\Phi_{ij}} = \begin{pmatrix} 0,23 & 0,58 & 1,3 \\ 0,3 & 3,52 & 2,13 \\ 2,67 & 0,95 & 0,95 \end{pmatrix}$$

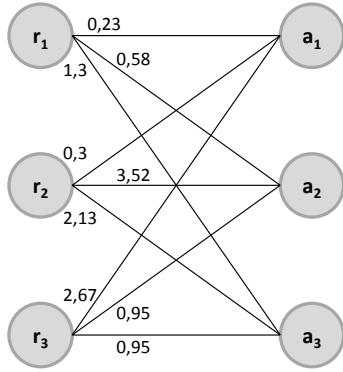


Figure 3. Resource-Task Relation Graph

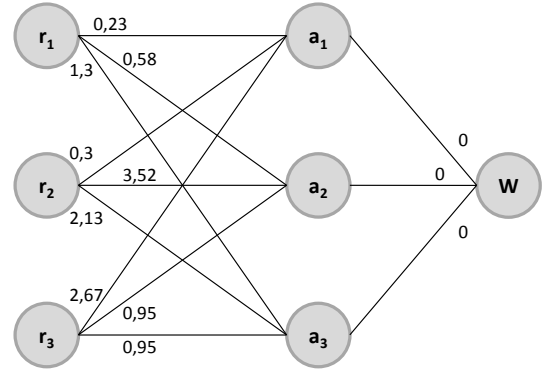


Figure 4. Extended Resource-Task Relation Graph

In order to calculate a MMT that solves the allocation problem according to the strategy chosen by one variation of Φ_{ij} , the following modification of Borůvka's algorithm is employed:

Algorithm: Minimum Matching Tree

Input: Connected, undirected graph $G = (V, E, f)$

Output: Set of edges E_{MMT} of a minimal matching tree.

1. $E_{MMT} := \emptyset, V := R \cup A \cup \{W\}$
 2. Add all edges adjacent to root W to E_{MMT}
 3. Remove vertex W from V
 4. While $|E_{MMT}| < |V| - 1$
 5. For $k = 1, \dots, m$ let e_k be an edge with minimum weight $f(e)$
 6. Remove adjacent vertices from V
 7. $E_{MMT} := E_{MMT} \cup \{e_1, \dots, e_m\}$
 8. E_{MMT} is a minimum matching tree
-

Note that R is a set of available resources, A is a set of tasks to be allocated. To transform G into a connected graph another (third type of) vertex W is introduced and connected to all tasks (edge weight is 0). Edges e_{ij} contained in E_{MMT} which connect vertices of R to vertices of A describe the allocation solution.

Due to its structure, the algorithms of Borůvka as well as its introduced modification are perfectly suitable for parallelization. Furthermore the introduced algorithm comes with good runtime characteristics; the upper bound, given in Landau notation (Landau, 1974; Knuth, 1997), is $O(|E| \log|V|)$:

In each phase of the algorithm edges with minimum weight are selected. This can be done by iterating over adjacent neighbors of each vertex within an upper bound of $O(|E| + |V|) = O(|E|)$ – because $|E| > |V|$ since G is connected. Let v_i be the number of vertices in phase i ; since each edge is connected to two vertices, $\{e_1, \dots, e_m\}$ does contain $\frac{v_i}{2}$ edges. In turn the number of vertices v_{i+1} in the next phase is reduced from v_i to a maximum of $\frac{v_i}{2}$. Therefore the number of phases can only be $O(\log|V|)$. ■

CASE STUDY

The evaluation of the allocation algorithm introduced above was conducted in cooperation with a major telecommunications service provider. Thus the algorithm has been evaluated for company internal ticket processing, which includes the maintenance of software systems, as well as the correction of minor software bugs. Currently the allocation of incoming tickets (tasks) is done manually, therefore one resource has to monitor all incoming tickets and suggest allocation to a resource. While ticket routing is of certain interest if accuracy of ticket assignment shall be evaluated (Sun, Tao, Yan, Anerousis and Chen, 2010), this is not considered in this case.

Resource	HR
Member A	(Member A, (K_{13} : 6), (S_2 : 3), (S_{11} : 5), (S_{21} : 4), (C_0 : 1), (C_1 : 2), (C_3 : 5), (C_5 : 6), (C_7 : 3))
Member B	(Member B, (K_1 : 6), (K_{39} : 3), (S_2 : 3), (S_{10} : 5), (S_{21} : 4), (C_1 : 1), (C_4 : 7), (C_5 : 3))
Member C	(Member C, (K_{12} : 6), (K_{39} : 3), (S_5 : 2), (S_{10} : 4), (C_6 : 3))
Member D	(Member D, (K_{12} : 6), (K_{23} : 5), (K_{33} : 2), (K_{39} : 3), (S_5 : 2), (C_6 : 3))
Member E	(Member E, (K_{12} : 2), (K_{32} : 3), (K_{39} : 3), (S_{10} : 4), (C_4 : 2))

Table 2

The evaluation is carried out for a team of five employees (displayed in Figure 2) handling tickets. Detailed properties of all five team members are shown in Table 2. Resource capabilities have been modeled according to historical ticket data and interview sessions. At first a description of capabilities according to given standard catalogues such as e-CF (CEN, 2010) had been tested, however, these turned out to be too coarse grained. Thus an individual capability model, which includes 8 competences, 44 skills and 33 knowledge instances, has been constructed. While competences mainly distinguish working areas, modeled skills basically serve as grouping mechanism of knowledge instances according to ticket classification. Knowledge instances primarily contain knowledge about modeling (e.g. about class diagrams), description and programming languages (such as JavaScript) and aspects of maintained systems. In order to analyze ticket requirements 480 tickets have been examined and assessed, in result a classification of tickets with according requirements profiles could be derived. Then, 120 supplementary tickets are used for allocation comparison between the current and suggested assignment strategies. Allocation is computed in discrete time steps; each resource does manage a task queue with a length of three, which means no more than three tasks can be allocated to a resource at one time step. If a task has not been completed until the next computed time step it still consumes a place in queue. The probability of task completion within time is 0,85 while the parameter δ is set to 4.

The allocation results are shown in Table 3. NoT is an abbreviation for *Number of Tasks executed*. It is recognizable that according to current state the resources A and B are utilized disproportionately (column NoT of Table 3), while allocation is more balanced in variant 2 (even in variant 1). According to definition of the distance measure Φ_{ij} introduced above, variant 1 computes a best fit based on capabilities, while variant 2 and 3 foster balanced allocation. However, in variant 3 only 82 of 120 tasks could be allocated at all. If this allocation strategy is chosen clearly δ has to be adjusted (values above 5 should be reasonable), since it cannot be acceptable not to solve incoming tickets. Through definition of δ a minimum level of capability match is enforced in variant 3. Variant 2 does lower differences between medium qualified resources; thereby a balanced learning curve is fostered. However, both variants result in longer times for ticket resolution (adjusted by lowering δ again). Differences in allocation are shown in Figure 5 by number of tasks. The standard deviation (number of allocated tasks per resource) is 23,6 (manual assignment), 20,8 (best fit, variation 1), 13,2 (balanced, variation 2) and 10,08 (minimum quality, variation 3).

The calculation of an optimization solution has quite good runtime characteristics; furthermore the algorithm can be parallelized by usage of adequate data structures; besides parallelization its runtime behavior can be further optimized by a combination with Prim's algorithm (Prim, 1957). In Table 4 a single allocation step is shown by matrices $M_{\Phi_{ij}}$ for each variant. Allocation chosen in this step is marked by underlined distances. In this case variant 2 and 3 are almost equal. Note that if distances between a task and different resources are equal, the given algorithm is not deterministic.

Resource	NoT	NoT – Variant 1	NoT – Variant 2	NoT – Variant 3
Member A	34	41	43	19
Member B	73	59	38	39
Member C	1	1	16	11
Member D	2	4	12	7
Member E	10	15	11	6

Table 3

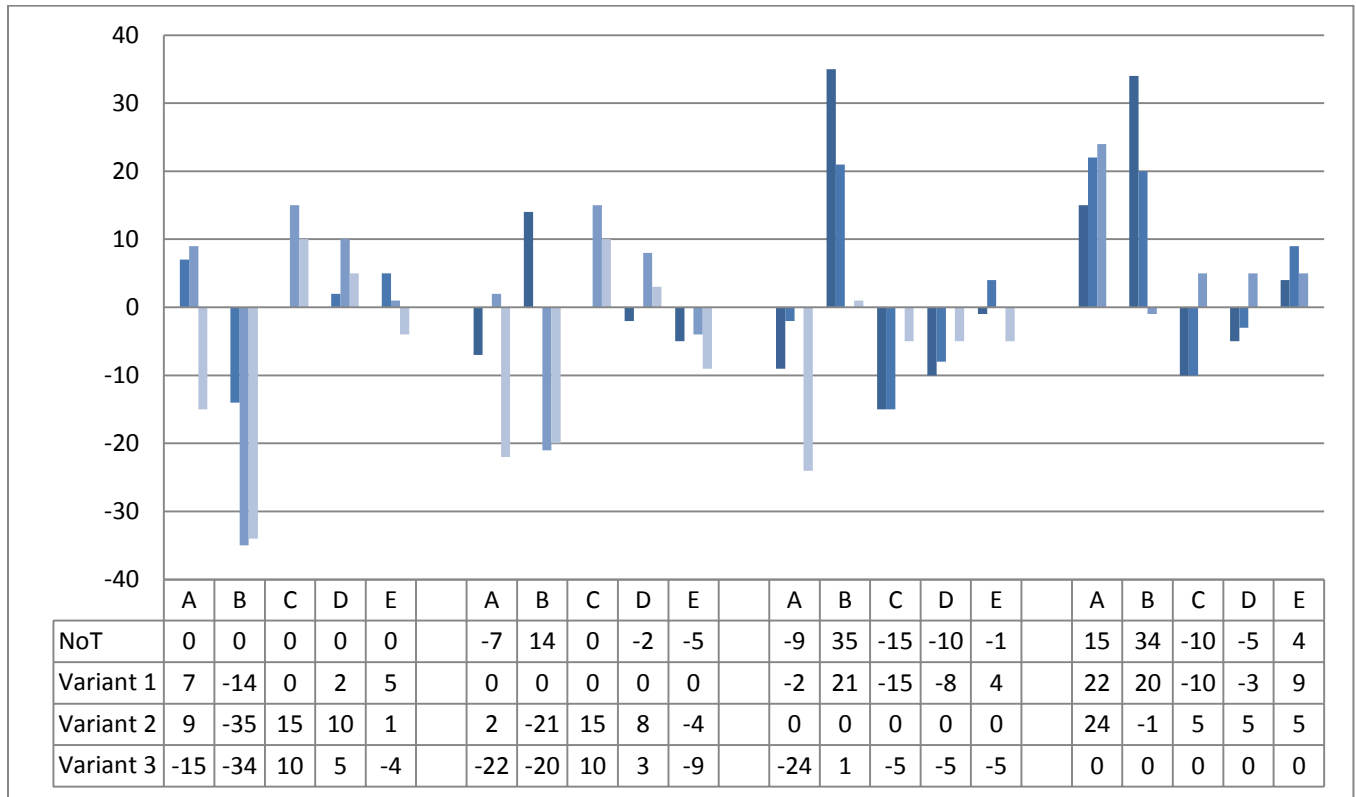


Figure 5. Allocation differences by distance measurement

Variant 1 – $M_{\Phi_{ij}}$	Variant 2 – $M_{\Phi_{ij}}$	Variant 3 – $M_{\Phi_{ij}}$
$\begin{pmatrix} 5 & \underline{2,65} & 5,39 & 5,39 & 5,74 \\ \underline{2,83} & 4,24 & 4,69 & 4,69 & 5,1 \\ 0 & 3,16 & 5,1 & 5,1 & 5,48 \\ 2 & \underline{1,41} & 3,74 & 3,74 & 4,24 \\ 2 & \underline{1,41} & 3,74 & 3,74 & 4,24 \end{pmatrix}$	$\begin{pmatrix} 1 & \underline{0,38} & 1 & 1 & 1 \\ \underline{0,35} & 1 & 1 & 1 & 1 \\ \underline{0,25} & 0,32 & 1 & 1 & 1 \\ 0,5 & 0,71 & \underline{0,27} & 0,27 & 1 \\ 0,5 & 0,71 & 0,27 & \underline{0,27} & 1 \end{pmatrix}$	$\begin{pmatrix} \infty & \underline{0,38} & \infty & \infty & \infty \\ \underline{0,35} & \infty & \infty & \infty & \infty \\ \underline{0,71} & \underline{0,32} & \infty & \infty & \infty \\ 0,5 & 0,71 & \underline{0,27} & 0,27 & \infty \\ 0,5 & 0,71 & 0,27 & \underline{0,27} & \infty \end{pmatrix}$

Table 4

CONCLUSION

In this article enhancements in capability based resource allocation of human resources could be demonstrated. Different strategies to allocate resources have been implemented by integration of different distance measures within one type of allocation problem definition. An advantage of this form of variation is that solutions of a single type of problem definition can be computed by the same algorithm. Hence the challenge to optimize the allocation according to different strategies is not the development of one algorithm per strategy but mapping the strategy to a distance measure within the problem definition – a much easier task to accomplish. Additionally the suggested optimization does scale in larger project scenarios because of the algorithm's runtime characteristics and possibility of parallelization.

In the presented case study different strategies are employed to achieve distinct objectives. In order to balance allocation amongst resources with similar capabilities, distance measures that promote challenging of resources are used. Besides a balanced allocation this does lead to implicit further education of resources with lower capability levels (capability gains can be computed as suggested by Hlaouittinun, Bonjour and Dulmet 2008; 2010). In consequence, peaks in capability levels are attenuated and the risk of losing capabilities (e.g. if a single resource leaves the company) is diminished.

Allocation optimization as suggested does operate on discrete points of time; therefore next steps of research are the evaluation of different strategies and algorithms. In addition to runtime aspects, heuristics for allocation and forecast of globally optimized solutions are of special interest. Furthermore, allocation of resources does also imply a gain in knowledge about process data, which in turn might be a security risk especially if allocation is capability driven only. Therefore another interesting field of research can be the evaluation of parameters which influence security (e.g. is capability based allocation compliant to security policies? Can a single allocation be compliant while the combination of several allocation decisions does result in security issues?).

REFERENCES

1. Adams M. (2009) The Resource Service, in *Modern Business Process Automation: YAWL and its Support Environment*, Springer, Berlin.
2. Borůvka O. (1960) Grundlagen der Gruppoid- und Gruppentheorie (Basics of gruppoid- and group theory), Vol. 46, VEB Deutscher Verlag der Wissenschaften.
3. CEN, European e-Competence Framework (2010) European e-Competence Framework 2.0 – A common European Framework for ICT Professionals in all industry sectors, European Commission.
4. CEN, European Qualification Framework (2008) The European Qualification Framework for Lifelong Learning (EQF).
5. Cook, C. and Churcher, N. (2005) Modelling and measuring Collaborative Software Engineering; *Proceedings of the Twenty-eighth Australasian conference on Computer Science*, Volume 38, Newcastle, Australia: Australian Computer Society, Inc., 267-276.
6. De Coi, J. L., Herder, E., Koesling, A. Lofi, C. Olmedilla, D., Papapetrou, O. and Sibershi, W. (2007) A model for competence gap analysis, in *WEBIST 2007*, Barcelona, Spain, INSTICC Press.
7. Du W. and Shan M. (1999) Enterprise Workflow Resource Management, *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, IEEE Computer Society.
8. Du W., Davis J., Huang Y, and Shan M.-C. (1999) Enterprise Workflow Resource Management, *Software Technology Laboratory HP Laboratories Palo Alto*, HPL-1999-8.
9. Fenz S., Ekelhart A. and Neubauer T. (2009) Business Process-Based Resource Importance Determination, *International Conference on Business Process Management*, 113-127.
10. Gutenberg, E. (1951) Grundlagen der Betriebswirtschaftslehre (Basics of economics, in German), Springer, Berlin, Heidelberg, New York, 24, 1983.
11. Harzallah M., Berio G. and Vernadat F. (2006) Analysis and modeling of individual competencies: toward better management of human resources, in *IEEE Transactions on Systems, man, and Cybernetics-PART A: Systems and Humans*, 36(1).

12. Hlaioittinun O., Bonjour E. and Dulmet M. (2008) A multidisciplinary team building method based on competency modelling in design project management, *International Journal of Management Science and Engineering Management*, Vol. 3 (2008) No. 3, 163-175, England, UK.
13. Hlaioittinun O., Bonjour E. and Dulmet M. (2010) Managing the Competencies of Team Members in Design Projects through Multi-period Task Assignment, *Collaborative Networks for a Sustainable World*, Springer Boston, 338-345.
14. Hollingsworth, D. (2005) The Workflow Reference Model: 10 Years On, Workflow Management Coalition (WfMC).
15. Hopcroft J. E. and Karp R. M. (1971) A $n^2/2$ algorithm for maximum matchings in bipartite, *In 12th Annual Symposium on Switching and Automata Theory*, 122-125.
16. HR-XML Consortium (2004) Competencies, Webpage: <http://www.hr-xml.org/>.
17. IEEE Learning Technology Standards Committee (2006) RCD: Draft standard for Reusable Competency Definitions (RCD), March 2006.
18. Kern W.(1988) Der Betrieb als Faktorkombination (Companies as combination of resources, in German), *In: Jacob, H. (Hrsg.): Allgemeine Betriebswirtschaftslehre*, 117-208.
19. Knuth, D. (1997) The Art of Computer Programming, Volume 1: Fundamental Algorithms; Asymptotic Representations, *Addison-Wesley, Third Edition*, 107-123.
20. Landau, E. (1974) Handbuch der Lehre von der Verteilung der Primzahlen, *Chelsea Publishing Company*, 3. edition XVIII.
21. Niemeyer, G (1977) Cybernetic System and Model Theory, System Dynamics (in German), Vahlen, Munich.
22. Oberweis, A. and Schuster T. (2010) A meta-model based approach to the description of resources and skills, *Proceedings of the Sixteenth Americas Conference on Information Systems*.
23. Object Management Group (2006) Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0.
24. Oliveira C., Lima R., Andre T. and Reijers H. (2009) Modeling and analyzing resource-constrained business processes, *IEEE International Conference on Systems, Man and Cybernetics (SMC 2009)*, 2824-2830.
25. OMG, Object Management Group (2010) Object Constraint Language” OMG Available Specification, Version 2.2.
26. Page B. and Kreutzer W. (2005) The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java, Shaker, Aachen.
27. Pfeffer J. and Salancik G. R. (2003) The external control of organizations: a resource dependence perspective, Stanford University Press.
28. Prim, R. C. (1957) Shortest connection networks and some generalisations, *in Bell System Technical Journal*, 36, 1389-1401.
29. Russell, N. and van der Aalst, W.M. (2008) Work Distribution and Resource Management in BPEL4People: Capabilities and Opportunities, *Proceedings of the 20th international conference on Advanced Information Systems Engineering*, Springer-Verlag, 94-108.
30. Russell, N., van der Aalst, W., ter Hofstede, A. & Edmond., D. (2005) Workflow resource patterns: Identification, representation and tool support, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, Springer, 216–232.
31. Staudt Lerner B., Ninan A.G., Osterweil L.J., Podorozhny R.M. (2000) Modeling and Managing Resource Utilization in Process, Workflow, and Activity Coordination, Technical Report, University of Massachusetts Amherst, MA.
32. Sun P., Tao S., Yan X., Anerousis N. and Chen Y. (2010) Content-Aware Resolution Sequence Mining for Ticket Routing, *Business Process Management*, Lecture Notes in Computer Science, 2010, Volume 6336/2010, 243-259.

33. van der Aalst, W. M. P. and Kumar, A. (2001) A Reference Model for Team-enabled Workflow Management Systems. *Data Knowl. Eng.*, 38, 3, 335-363.
34. van der Aalst, W. M. P., Kumar, A. und Verbeek, H.M.W. (2003) Organizational modeling in UML and XML in the context of workflow systems; *Proceedings of the 2003 ACM symposium on Applied computing*, Melbourne, Florida, S. 603-608.
35. Winterton, J. and Delamare- Le Deist, F. and Stringfellow, E. (2005) Typology of knowledge, skills and competences: clarification of the concept and prototype; Cedefop 2005.
36. zur Mühlen, M. (1999) Evaluation of Workflow Management Systems Using Meta Models, R. Sprague (Ed.). *Proceedings of the 32nd Hawaii international Conference on System Sciences*.
37. zur Mühlen, M. (2004) Organizational Management in Workflow Applications - Issues and Perspectives, *Information Technology and Management*, 5, 271-291.